

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

2D arkáda pre platformu Android
2D Arcade Game for Android Platform

Zadání bakalářské práce

Student:	Křištof Kubík
Studijní program:	B2647 Informační a komunikační technologie
Studijní obor:	2612R025 Informatika a výpočetní technika
Téma:	2D arkáda pro platformu Android 2D Arcade Game for Android Platform

Zásady pro vypracování:

Cílem práce je vytvořit arkádu, zaměřenou primárně na platformu Android. Obtížnost hry se bude přizpůsobovat schopnostem hráče, protihráčem bude zejména umělá inteligence (AI) s prvky kooperativního chování, implementovaná v rámci této práce.

1. Navrhněte téma hry a srovnajte obdobné existující hry na mobilní platformě Android.
2. Definujte scénář hry a způsob(-y) ovládání hry.
3. Zaměřte se na oblast algoritmů pro kooperaci postav/jednotek protivníka (NPC), vyberte jeden či více algoritmů, vhodných pro mobilní platformu implementujte je (s případnými změnami), včetně možnosti (automatického) nastavení schopností tohoto protihráče.
4. Zpracujte herní logiku včetně základní AI protihráče a hru implementujte.
5. Výsledné řešení otestujte a - bude-li to možné - nabídněte hráčské komunitě.

Seznam doporučené odborné literatury:

1. J. F. DiMarzio: Practical Android 4 Games Development. Apress, 2011, ISBN: 978-1-4302-4029-7
2. Meier, R. Professional Android 2 Application Development. Wrox Press Ltd., 2010. ISBN: 978-0-47056-552-0.
3. Scerri, P., Johnson, L., Pynadath, D. V., Rosenbloom, P., Schurr, N., Si, M., & Tambe, M. (2002). Getting Robots, Agents and People to Cooperate: An Initial Report.
4. Yannakakis, G. N., & Hallam, J. (2004). Evolving opponents for interesting interactive computer games. From animals to animats, 8, 499-508.
5. Jie, J., Yang, K., & Haihui, S. (2011, October). The Application of AI for the Non Player Character in Computer Games. In Computational and Information Sciences (ICCIS), 2011 International Conference on (pp. 1049-1050). IEEE.
6. Mike Preuss et al.: Towards Intelligent Team Composition and Maneuvering in Real-Time Strategy Games, IEEE Transactions on Computational Intelligence and AI in Games. 2. Vol. 2. 2010. pp. 82-98.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Vyhlásenie študenta

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

Dňa: 7.5.2014


.....

podpis študenta

Pod'akovanie

Rád by som pod'akoval Ing. Pavlovi Moravcovi, PhD. za odbornú pomoc a konzultáciu pri vytváraní tejto diplomovej práce

Abstrakt

Táto bakalárska práca sa zaoberá vývojom 2D arkádovej hry pre operačný systém Android. Podstatou bolo preskúmať algoritmy pre vyhľadávanie a kooperáciu, vybrať ich vhodnú kombináciu pre umelú inteligenciu a implementovať ich. Úvodná časť práce sa zaoberá použitými technológiami, rozborom hier a predstavení umelej inteligencie. V praktickej časti vyberám vhodné algoritmy, analyzujem hru a popisujem problémy pri implementácii.

Kľúčové slová:

2D hra, Arkáda, Pathfinding, A*, Depth-first search, Pursuit-evasion

Abstract

This bachelor thesis deals with the development of 2D arcade game for Android operating system. The main idea was to examine algorithms for searching and co-op, choose the appropriate combination of artificial intelligence and implement them. The introductory part deals with the technology used, analysis of games and performances of artificial intelligence. In the practical part selects appropriate algorithms, analyze the game and describe implementations problems.

Keywords:

2D game, Arcade, Pathfinding, A*, Depth-first search, Pursuit-evasion

Zoznam použitých skratiek a symbolov

AI	- Artificial Intelligence
API	- Application Programming Interface
2D	- Two Dimensional
3D	- Three Dimensional
GUI	- Graphical User Interface
IQ	- Intelligence Quotient

Obsah

Úvod	1
1 Použité technológie	2
1.1 Platforma Android	2
1.1.1 História	2
1.1.2 Fragmentácia	2
1.1.3 Android API	3
1.1.4 Google Play	4
1.1.5 Google Play Store	4
1.1.6 Google Play Music	4
1.1.7 Google Play Movies & TV	4
1.1.8 Google Play Books	4
1.2 Framework	5
2 Hry	6
2.1 Adventúry	6
2.2 Akčné	7
2.3 Arkády	7
2.4 Detské	8
2.5 RPG	8
2.6 Simulátory	8
2.7 Stratégie	8
2.8 Casual hry	9
2.9 Existujúce arkádové hry	9
2.9.1 Vogelstein 2D	9
2.9.2 Bomberman classic	10
2.9.3 Pac-Man	10
3 Umelá inteligencia	12
3.1 Druhy porozumenia	13
3.2 Silná umelá inteligencia	13
3.3 Slabá umelá inteligencia	14

3.4	Vyhľadávacie algoritmy (PathFinding).....	15
3.5	Informované metódy vyhľadávania.....	15
3.6	Stavový priestor.....	16
3.7	Neinformované metódy vyhľadávania	17
4	Výber vhodného algoritmu pre pathfinding	18
4.1	Vyhľadávací algoritmus depth-first.....	18
4.2	Vyhľadávací algoritmus A*	19
5	Výber vhodného algoritmu pre kooperáciu	21
5.1	Predator-prey	21
5.2	Pursuit-evasion	21
6	Analýza.....	23
6.1	Základný koncept	23
6.2	Možnosti ovládania	24
6.2.1	Ovládanie pomocou naklápania.....	24
6.2.2	Ovládanie pomocou dotyku.....	24
6.2.3	Ovládanie externým ovládačom	25
6.3	Návrh herných modulov	25
7	Implementácia	27
7.1	Ovládanie.....	27
7.2	Hlavná ponuka.....	28
7.3	Mapa.....	28
7.4	Pohyb.....	29
7.5	Testovanie kolízií	30
7.6	Vyhľadávanie cesty	31
7.7	Nastavovanie schopností jednotiek	32
8	Testovanie	33
8.1	Testovanie na reálnom zariadení	33
	Záver.....	35
	Literatúra	36
	Obsah priloženého CD	37

Úvod

Trh s mobilnými aplikáciami by sa v súčasnosti dal jednoducho charakterizovať ako atraktívny. Keďže neustále rastie počet majiteľov mobilných telefónov s operačným systémom, ktorý umožňuje inštalovať aplikácie. Takých systémov je viacero a nové pribúdajú. Väčšina obsahuje obchod s aplikáciami, poskytujúci užívateľovi rýchly prístup k už existujúcim aplikáciám. V systéme Android je takáto distribúcia už dlhodobo zaužívaná a veľmi obľúbená. Jediným kliknutím je možné si aplikáciu stiahnuť, poprípade zaplatiť. Ceny sa rádovo pohybujú na veľmi nízkych hodnotách a teda užívatelia nemajú problém si takéto aplikácie zaobstaráť.

Cieľom mojej práce je popísať vývoj arkádovej hry pre platformu Android. Práca sa vo svojej podstate zaoberá umelou inteligenciou. V prvej časti práce stručne zhrňujem históriu a problematiku systému Android. V ďalšej kapitole som sa pokúsil oboznámiť s umelou inteligenciou použiteľnú pre arkádovú hru na platforme Android. Zameriavam sa na oblasť vyhľadávania cesty(pathfinding). Analyzujem možnosti ovládania a navrhujem základný koncept hry a jej moduly. Popisujem implementáciu a zvolenú kombináciu algoritmov. Zaver už približuje celkové zhodnotenie práce.

1 Použité technológie

Popis použitých technológií vhodných na tvorbu 2D hry.

1.1 Platforma Android

V tejto kapitole som čerpal z[1].

Android je otvorený operačný systém, platforma je postavená na Linuxovom jadre, určená pre mobilné telefóny a zariadenia. Vývoj Androidu prebieha pod open source licenciou. Za vznikom tejto platformy stojí firma Android Inc. Pod pojmom otvorený mobilný systém rozumieme mobilný systém, ktorý nám dovoľuje inštalovať aplikácie, ale na rozdiel od ostatných mobilných systémov sa líši tým, že máme možnosť upraviť časti systému podľa svojich predstáv.

1.1.1 História

Android Inc. spoločnosť bola založená v roku 2003 a v roku 2005 ju odkúpil Google Inc. Firma pod vedením Andyho Rubina vyvinula platformu na baze Linuxového jadra a získala patenty v oblasti mobilných technológií. Google týmto krokom dala najavo že mieni vstúpiť na pole "smartfonov" a chystá vídať vlastný telefón.

5. novembra 2007 bola vytvorená tzv. "Open Handset Alliance", v histórii Google asi najdôležitejší krok. Túto alianciu predstavujú poprední výrobcovia mobilných telefónov a cieľom bolo vyvinúť potrebný štandard. V ten istý deň ohlásil Google prvú verziu operačného systému Android.

Prvý telefón s Androidom bol uvedený na trh o rok neskôr, jeho výrobcom bol HTC. Smartphone s Androidom 1.0 mal na trhu malý 0,5% podiel. Prelom na trhu prišiel v máji 2010, pri predstavení Android 2.2(Froyo). V roku 2012 už každý druhý smartfon obsahuje Android.

Android je open-source, z tohto dôvodu výrobcovia telefónov môžu ponúknuť podstatne lacnejšie ceny telefónov ako tomu bolo v minulosti. Popularita toho systému spočíva taktiež v jednoduchosti ovládania tzv. "user-friendly" pre koncových užívateľov.

1.1.2 Fragmentácia

Tak ako každý systém aj Android má svoje nevýhody. Výrobcovia smartfonov upravujú systém a prinášajú pre užívateľov vlastne nadstavby užívateľského rozhrania. Tieto nadstavby zaostávajú za najnovšími verziami Androidu a tak užívateľom nie je umožnené prejsť na vyššiu

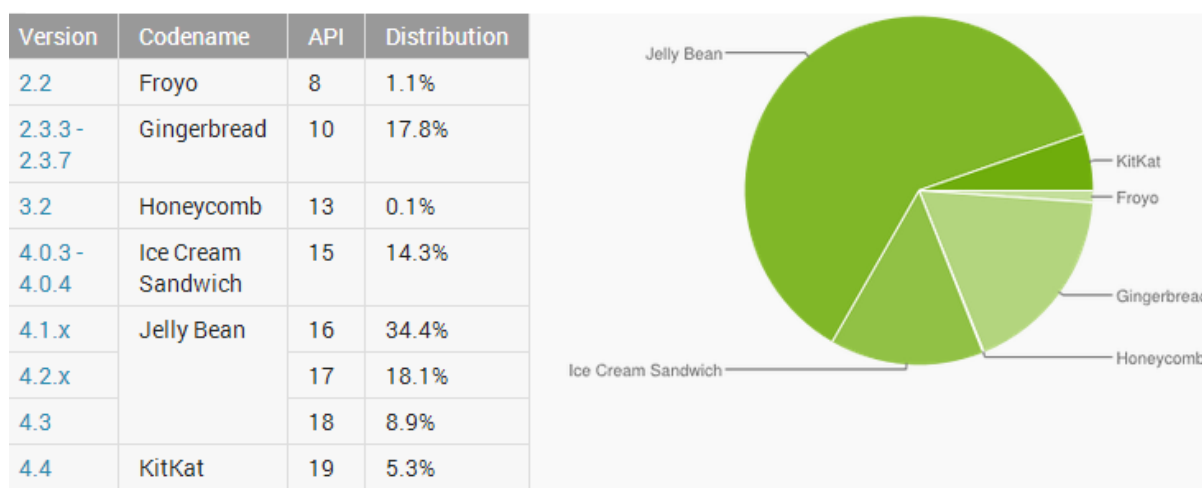
verziu. Pre elimináciu tohto javu musia vývojári spraviť opatrenia ktoré zabezpečia stabilný beh aplikácií na všetkých verziách Androidu. V prípade aby aplikácia určená pre novšie verzie Androidu fungovala aj na starších je nutne písať oddelený kód pre rôzne verzie.

Fragmentácia pri vývoji hier nie je až tak výrazná pretože nie je potreba používať veľa Adroid API. Hardwaerova fragmentácia je už väčší problém a vyskytuje sa i na iných platformách. Množstvo zariadení ma z veľkej časti odlišné parametre, veľkosť, rozlíšenie displeja, veľkosť RAM pamäti, výkon procesoru. Je potrebné počítať s menším výpočtovým výkonom pri starších alebo tzv. "low-end" zariadeniach.

1.1.3 Android API

Pre vývojára je jedným z prvých rozhodnutí pre akú cieľovú verziu operačného systému Android sa rozhodne aplikáciu implementovať. Pri zvolení najaktuálnejšej verzie je možné prísť o potencionálnych zákazníkov ktorý so staršími verziami systémov nemôžu predmetnú aplikáciu používať.

Google pravidelne zverejňuje aktuálne zastúpenie jednotlivých verzií Android systémov, ktoré pristupujú na službu Google Play. S týmito údajmi je pre vývojára jednoduchšie sa rozhodnúť pre akú verziu ma svoju aplikáciu zvoliť s ohľadom na prípadný zisk. Mojou voľbou je verzia 4.1.x (Jelly Bean) Api16.



Obrázok 1: Verzie Androidu pristupujúce do služby Google Play k 1.4.2014[1]

1.1.4 Google Play

Google Play je on-line distribučná služba, ktorá vznikla 6.3.2012 spojením služieb Google Music a Android Market. V súčasnosti Google Play poskytuje niekoľko druhov digitálneho obsahu, ku ktorým je možné pristupovať z viacerých platforiem(PC, mobilný telefón, Google TV).

Hlavným cieľom Google Play je distribúcia aplikácií pre mobilné telefóny a tablety s operačným systémom Android, táto časť sa nazýva Google Play Store. Ďalšia oblasť je on-line distribúcia hudby s názvom Google Play Music. Tretia v poradí nesie názov Google Play Movies & TV a je zameraná na distribúciu filmov. Posledná časť distribuuje elektronické knihy a volá sa Google Play Books.

1.1.5 Google Play Store

Google Play Store je priamym nástupcom Android Marketu ktorý bol otvorený pre vývojárov tretích strán a ich aplikácií. K dispozícii sú bezplatné i platené aplikácie nahrávané na Play Store vývojármi. Prostredníctvom Play Store môže vývojár ponúkať aktualizácie aplikácií a získavať spätnú väzbu od užívateľov.

Sťahovanie a prípadné zakúpenie aplikácie je možné prostredníctvom mobilného telefónu alebo webového rozhrania, pri splnení podmienok.

1.1.6 Google Play Music

Táto služba bola spustená 16.11.2011, ide o on-line hudobnú knižnicu, v ktorej je možné zakúpiť si jednotlivé skladby alebo albumy. Súčasťou služby je priestor pre uloženie vlastných skladieb. Do tejto knižnice je možné nahráť až 20 000 skladieb, ktoré sa užívateľovi sprístupnia pre všetky zariadenia s operačným systémom Android.

1.1.7 Google Play Movies & TV

Služba sa zaoberá sprostredkovaním filmov a videí. Po registrácii je Google Play Movies & TV dostupný z webového prehliadača.

1.1.8 Google Play Books

Google Play Books je zameraná na distribúciu elektronických kníh. V databáze sa súčasne nachádza viac ako 4 milióny rôznych kníh. Niektoré si je možné zapožičať a vytvárať si vlastnú virtuálnu knižnicu dostupnú z viacerých zariadení súčasne.

1.2 Framework

Pre zjednodušenie implementácie hry som zvolil herný framework andengine[2,3]. Tento framework uľahčuje vývoj hry a často používané operácie, ktoré by som musel pri každom použití volať. Namiesto toho komunikácia prebieha iba s frameworkom, ktorý je dostatočne vyladený a zariadi všetko potrebné. Framework sa delí na moduly, ktoré som v práci používal:

- **Správa obrazovky:** tento modul sa stará o vytváranie View, do ktorého sa vykresľuje obraz hry. Taktiež rieši akcie ako sú: pozastavenie, minimalizovanie, obnovenie hry.
- **Vstup:** pracuje sa tu so vstupnými senzormi, v mojom prípade využívam dotykový display (môže sa použiť klávesnica, akcelerometer pre prípadné ovládanie nakláňaním zariadenia). Tieto údaje sa prevedú do použiteľného formátu s ktorým sa dá ľahko pracovať.
- **Grafika:** modul ktorý sa stará o vykreslenie grafiky na obrazovku či v podobe úvodného loga (Splash screen) alebo samotnej hry (vykreslenie mapy, pohybov a pod.)
- **Hra:** spája všetky moduly pomocou kompozície a tým pádom umožňuje jednoduché vytvorenie hry jej rozšírením.

2 Hry

V celej tejto kapitole som čerpal z[4,5].

Každá hra sa dá zaradiť podľa rôznych kritérií, či už prostredím, kde sa odohráva, pohľadom kamery (pohľad z prvej/tretej osoby).

Najdôležitejším faktorom zaradovania hier je žánr.

Žánre hier:

- Adventúry
- Akčné
- Arkády
- Detské
- Rpg
- Simulátory
- Stratégie
- Casual hry

2.1 Adventúry

Adventúra je žánr hry založený na príbehu. Hlavný hrdina rieši úlohy a tak postupuje príbehom k cieľu. Kladie sa tu veľký dôraz na riešenie rôznych logických hádanok.

Štruktúrovo je u adventúry vytýčený tzv. 'Main Quest' (hlavná úloha), ktorý obsahuje rôzne 'Quests' (úlohy) ktoré hráč plní s využitím svojej fantázie a logického myslenia.

Najviac sa adventúram darilo v 90. rokoch 20. storočia. Zo začiatku sa príkazy hrdinovi zadávali cez príkazový riadok. Neskôr sa pristúpilo ku grafickému spracovaniu tzv. 'Point and click' a k príkazom zobrazeným na ovládacom paneli.

2.2 Akčné

Akčná hra je označenie pre hru, ktorou hlavnou náplňou je eliminácia cieľov pomocou bojových techník. Tieto hry väčšinou virtuálne znázorňujú dynamický boj, násilné sekvencie a škálu zbraňového arzenálu. V niektorých krajinách musí byť hra takéhoto žánru psychológmi skontrolovaná, prípadne cenzurovaná, aby nenabádala hráča k mimoriadne násilníckym sklonom.

V takýchto hrách sa objavujú vždy po určitom počte misií alebo epizód tzv. bossovia. Najnovšie 3D akčné hry sú obvykle náročné na hardware a často sa vyznačujú veľmi dobrou umelou inteligenciou nepriateľov, pokročilým grafickým spracovaním a vizuálnymi efektmi. Hráč najčastejšie ovláda jednu postavu (alebo menšiu skupinu), zbraň alebo vozidlo a niekedy si môže voľiť náročnosť (napríklad: začiatočník, pokročilý, expert).

2.3 Arkády

Arkáda je žáner počítačovej hry, založený na jednoduchom a nápaditom koncepte. Arkády sa najčastejšie hrajú na kolá s postupne rastúcou náročnosťou, niektoré hry bývajú obmedzené časom. Podľa námetu rozhoduje hlavne postreh, rýchlosť pohybu prstov po klávesnici a jeho zautomatizovanie, rýchlosť logického myslenia, predvídanie, trpezlivosť, koncentrácia, rýchlosť prevedenia nejakej akcie atď. Arkádovými rysmi sa označujú koncepty, ktoré spravidla pomocou zmäkčenia fyzikálnych zákonov prispievajú k väčšej "zbesilosti" a jednoduchosti hrania hry, ktorá sama o sebe arkádou nie je. V arkádach záleží ako na šikovnosti, tak aj na dôvtipe.

V arkádach záleží taktiež na šikovnosti a dôvtipu, podľa množstva vydávaných hier na rovnakom princípe sa začali deliť do kategórií podľa spoločných znakov. V Arkádach sa stala veľmi populárna hra pre dva a viac hráčov na jednom počítači a také hra v módu 'Split Screen'.

Arkády delíme na:

- Bojové hry
- Plošinové hry
- Športové hry
- Závodné hry
- Logické hry

2.4 Detské

Detské hry z pravidla nemajú žiadne obmedzenie čo sa týka žánru. Jediným pravidlom je jednoduchosť.

2.5 RPG

Skratka RPG ang. Role Playing Game (Rolová hra) je typ hry, pri ktorej sa hráči vžívajú do úloh imaginárnych postáv, umiestnených do prostredia a snažia sa za tieto postavy konať.

Častokrát daná postava predstavuje virtuálnu reprezentáciu hráča. Hra umožňuje zdokonaľovanie schopnosti postavy a jej vývin. Zmyslom RPG je vytvoriť pre hráčov virtuálny svet, v ktorom môžu získať skutočné/neskutočné schopnosti a zažiť dobrodružstvá.

2.6 Simulátory

Tento žáner hier je založený na presnej simulácii (spoločenských, ekonomických, fyzikálnych atd.) aspektov ktoré majú predstavovať skutočný svet. Hráč má k dispozícii rozsiahle nastavenia simulovanej skutočnosti, čo zvyrazňuje autenticitu hry.

Najčastejšie sa simulátory zameriavajú na simulovanie závodov, športu, lietadiel, spoločenského života.

2.7 Stratégie

Základom tohto žánru je stavba základne, vyrábanie jednotiek a ich vylepšovanie resp. boj proti nepriateľom. Hráč dostane na začiatku každej misie inštrukcie, ktoré splnením vedú k víťazstvu. Najčastejšie sa jedná o zničenie nepriateľských budov/jednotiek, zabratie územia alebo ťažba surovín.

Stratégie sa môžu odohrávať v reálnom čase alebo na kolá, tzv. ťahové stratégie.

2.8 Casual hry

Veľmi populárny žáner určených pre nenáročných hráčov - nemajú moc skúsenosti s hraním hier. Základným znakom je veľká návykovosť, ktorá dokáže hráčov zamestnať na hodiny - hra nemá žiaden cieľ. Casual hry sa hrajú zväčša prostredníctvom internetového prehliadača.

Popularita tohto žánru spočíva v nekonečnej variabilite zamerania každej danej hry (každá hra môže byť originálna).

2.9 Existujúce arkádové hry

V tejto časti popisujem existujúce arkádové hry ktoré je možné kedykoľvek stiahnuť z Google Play. Každá z týchto hier má rovnaké aj odlišné prvky.

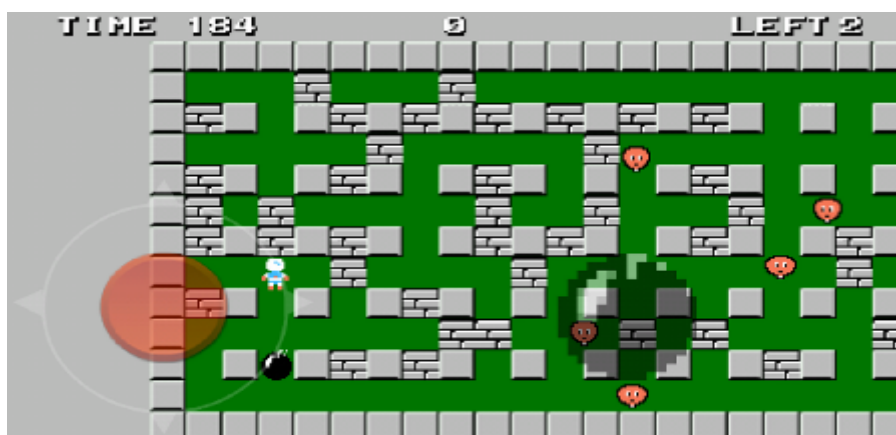
2.9.1 Vogelstein 2D



Obrázok 2:Obrázovka z hry Vogelstein 2D [5]

Hra Vogelstein 2D je inšpirovaná klasickou hrou Wolfenstein 3D, je orientovaná do prostredia druhej svetovej vojny. V príbehu sa ako hráč ocitnete v úlohe väzňa ktorý sa snaží uniknúť zo zámku Vogelstein. Počas 20 levelov ste konfrontovaný s vlnami nepriateľov. Nepriatelia sú rozdelení od najslabších útočných psov až po najsilnejších SS dôstojníkov. Sú tu k dispozícii 3 typy zbraní a 3 obtiažnosti (ľahká, stredná, ťažká).

2.9.2 Bomberman classic



Obrázok 3:Obrázovka z hry Bomberman classic[5]

Bomberman je ďalšou klasickou hrou dostupnou pre platformu Android. Hra je rozdelená do 8 svetov, kde každý svet má ďalších 8 levelov. Každý level je reprezentovaný ako labyrint kde sa pohybujú nepriatelia. Pomocou bômb, musí hráč zničiť všetkých nepriateľov, ak sa mu to podarí odkryje sa mu vchod do ďalšieho levelu.

2.9.3 Pac-Man



Obrázok 4:Obrázovka z hry Pac-Man[5]

Pac-Man je japonská arkádová hra vyvinutá spoločnosťou Namco. V Japonsku sa táto hra rýchlo stala populárnou a symbolom počítačových hier. Vo svojej dobe (1980) to bol priekopník a vytvoril úplne nový typ hry. Hráč ovláda Pac-Mana po mape a snaží sa zjesť všetky bodky, ak sa mu to podarí postupuje do ďalšieho levelu. Štyria nepriatelia (duchovia) majú každý vlastnú logiku ktorou sa pohybujú po mape. Ak sa nepriateľ dotkne Pac-Mana odpočíta sa život, po klesnutí životov na 0 hra končí.

3 Umelá inteligencia

"Přirozená intelligence bude umělou brzy překonána. Přirozenou blbost však umělá nemůže nahradit nikdy." Jára Cimrman

V celej tejto kapitole čerpám informácie z [6,7].

Inteligencia je vlastnosť živých organizmov, ktorá im dáva v prírode mimoriadne postavenie. Vznikla a vyvíjala sa v priebehu dlhého vývoja. Dnes umožňuje niektorým živým organizmom efektívne reagovať na zložité prejavy prostredia a aktívne ju využívať vo svoj prospech, k dosiahnutiu svojich cieľov.

Súbežne s rozvojom techniky si ľudia kladú otázku, či i pri umelo vytvorených systémoch možno dosahovať reakcie a celkového chovanie, ktorému by sme u živých organizmov dali prívlastok inteligentní. Tieto otázky si ľudia kládli už dávno pred vynájdením elektronických počítačov a snažili sa nimi zaoberať v rámci vtedajšej filozofie a neskôr aj psychológie.

Až niektoré významné výsledky v matematickej logike a v teórii algoritmov dosiahnutých v 30. rokoch dvadsiateho storočia sú spojené predovšetkým s menom nemeckého matematika Gödela a hlavne tak nástup a veľmi rýchly rozmach výpočtovej techniky po druhej svetovej vojne sa stali skutočnými katalyzátormi úsilia napodobniť intelektuálne schopnosti najvyspelejšieho živého tvora - človeka. Postupne sú navrhované a experimentálne overované metódy, algoritmy umožňujúce určité hľadisko inteligentného chovania napodobňovať. K tomu sú používané techniky vychádzajúce z detailnej analýzy činnosti živých organizmov na úrovni biologickej, tak aj techniky vychádzajúce z matematickej abstrakcie mentálnych procesov ľudského mozgu na úrovni psychologickej a kognitívnej.

Pojem "inteligencia" u živých organizmov nie je presne ohraničená. Existujú síce metódy merania inteligencie (test IQ), ale každá z nich má svoje hranice a nikto nemôže tvrdiť, že niektorá je dokonale objektívna. Z tohto dôvodu pojem "umelej inteligencie" nebol doteraz presne a jednoznačne definovaný.

3.1 Druhy porozumenia

Level implementácie inteligencie pomocou umelého média závisí v princípe na tom do akej miery sa podarí vytvoriť kľúčový faktor ľudského rozumu. Rozum - chápanie, tak isto ako aj veľa ďalších kvalít je možné charakterizovať v zmysle silnej alebo slabej miery. Ak hovoríme o myšlienke, systéme že je slabý, máme tým namysli že je obecný. Naopak silná myšlienka je vo svojej podstate špecifická. Systém GPS, ktorý na princípe matematickej logiky rieši obecné problémy, bol klasifikovaný ako slabý.

S ohľadom na to aké porozumenie sa nám podarí vytvoriť delíme umelú inteligenciu:

1. Slabú umelú inteligenciu dosiahneme ak vytvoríme slabé porozumenie. Slabé porozumenie chápeme ako porozumenie že systém na správne vstupne podnety vykazuje korešpondujúce reakcie.
2. Silnú umelú inteligenciu dosiahneme, ak vytvoríme silné porozumenie. Silné porozumenie chápeme ako porozumenie že systém bude disponovať pocitom chápaným takým, akým disponuje ľudská myseľ

3.2 Silná umelá inteligencia

Silná umelá inteligencia(Strong AI) - je inteligencia, o ktorej môžeme povedať, že sa vyrovnáva a možno v niektorých ohľadoch prekonáva ľudskú inteligenciu. Je schopná vykonávať ktorúkoľvek intelektuálne založenú úlohu tak ako človek. Je známa najmä z rôznych sci-fi filmov alebo kníh. Silná umelá inteligencia je vo všeobecnosti dosť často označovaná ako samotná umelá inteligencia vôbec. Medzi oporné body patria:

- vedomosť o vlastnej existencii - samostatná jednotka ktorá si je vedomá samej seba
- vnímavosť - schopnosť vnímať okolie a popípade reagovať.
- vedomie - samostatné subjektívne myslenie
- strategické myslenie
- komunikácia - schopnosť komunikovať v zrozumiteľnej reči
- riešenie neznámych úkolov
- plánovať

3.3 Slabá umelá inteligencia

Slabá umelá inteligencia(weak AI) - je inteligencia, ktorej úlohou nie je vyrovnat' sa ľudskej, ako to je u silnej umelej inteligencie. Neobsahuje prvky ako myseľ, vedomosť vlastnej existencie. Hlavným účelom je sústredenie sa na riešenie malých a konkrétnych činnosti. Ide o stroj, ktorý sa snaží o nájdenie nových riešení, stratégií, ako riešiť vzniknutý problém. Na riešenie týchto problémov využíva vopred nadefinované algoritmy. Pri spracovávaní nových informácií sa ich snaží čo najefektívnejšie použiť. Slabá inteligencia môže mať taktiež rovnaké črty ako silná inteligencia ale nemusí ich vedieť použiť v správny čas na správnom mieste a teda ich efektívnu kombináciu. V súčasnosti sa výskum čoraz viac zameriava na slabú inteligenciu, ktorá má veľký potenciál pre jej širokospektrálne použitie. Oporné body slabej inteligencie:

- ovládanie, hranie hier
- ovládanie robotov
- plánovanie
- riadenie
- navigácia
- triedenie
- analýza
- predpovedanie
- spracovanie dát
- kombinatorika

3.4 Vyhľadávacie algoritmy (PathFinding)

Základom umelej inteligencie v priestore sú vyhľadávacie algoritmy. I keď v súčasnosti sú dnešné osobné počítače alebo mobilné telefóny na úrovni starších super počítačov, čiže majú dostatok výpočtového výkonu, stále sa vývojári držia pravidla neplytvat' výkonom na neefektívne riešenia. Z tohto dôvodu by u vyhľadávacieho algoritmu bol nezmysel riešiť všetky možné cesty z ktorých by sa vyberala ta najkratšia. Preto vznikli rôzne techniky ako tomu predísť a vybrať kompromis medzi časovou zložitou (minimálny a maximálny čas potrebný na vykonanie algoritmu), priestorovou zložitou (minimálny a maximálny počet riešení), kvalitou získaných výsledkov (pomer medzi pravdivosťou a správnosťou informácií).

Problém ale spočíva akým spôsobom je možné nájsť najkratšiu cestu, pričom prehľadávaný priestor nemusí byť konečný. V tomto prípade ale vieme určité stavy a pozície, pre ktoré chceme nájsť najkratšiu cestu. Ak je daný priestor príliš veľký, môže nastať problém s nedostatkom pamäte, pretože algoritmus si musí pamätať už vyhodnotené stavy. Ďalej môže nastať prípad, kde bude priestor tak zložitý, že zvolený algoritmus nebude schopný správne nájsť cestu a k cieľu vôbec doraziť. Z tohto dôvodu vzniklo viacero postupov ako hľadať najkratšiu cestu v priestore. Je nemožné zvoliť jeden algoritmus, ktorý by sa hodil pre každý prípad, keďže vždy sú podmienky kladené na vstup iné. Spôsoby vyhľadávania sa delia na informované a neinformované techniky. Hlavný rozdiel je v tom, aké majú vstupné informácie o stavovom priestore.

3.5 Informované metódy vyhľadávania

Základom tejto metódy je, že pozná stavový priestor. Tým pádom získava výhodu, pretože vie lepšie odhadnúť, aká je veľká vzdialenosť od aktuálneho stavu. Tento odhad predstavuje heuristická funkcia $h(n)$. Čím menšiu $h(n)$ hodnotu majú stavy n , prevedie sa cesta týmito stavmi. Z toho vyplýva že lepšie ohodnotené stavy heuristickou metódou, majú menšie nároky na pamäť a sú teda efektívnejšie a rýchlejšie. Heuristická informácia sa práve preto využíva na zvýšenie efektívnosti hľadania. Je to spôsob, pri ktorom algoritmus vyberá medzi viacerými riešeniami, alternatívami. Podmienkou je, ale aby sa vždy čo najefektívnejšie dostal k cieľu. Medzi tieto metódy vyhľadávania patria:

-
- Best-first search: vyberá vhodný uzol k expanzii. Takýto uzol vedie prehľadávanie k optimálnemu riešeniu. Ak funguje heuristicky algoritmus dobre, minimalizuje prehľadávané časti stavového priestoru, ktoré nevedú k optimálnemu riešeniu.
Greedy search , A* search;
 - Local-search: lokálna metóda prehľadávania
Hill-climbing, Simulated annealing

3.6 Stavový priestor

Stavový priestor je štvorica (N, A, S, G) popisujúca konfiguráciu stavov.

- N - stavová množina
- A - prechod medzi stavmi
- S - množina s počiatočnými stavmi
- G - množina s cieľovými stavmi

V jednoduchosti si ho môžeme predstaviť ako graf, ktorý ma jednotlivé stavy ohodnotene, taktiež si ho môžeme predstaviť ako sieť poskladanú zo štvorcov a každý nesie informáciu o stave. Na prechod stavového priestoru slúžia vyhľadávacie algoritmy.

Sú definované počiatočné a finálne stavy. Medzi akýmkoľvek stavmi je možné aplikovať určité prechody, tie je možné prechádzať, tak nám vznikne daný stavový priestor. V princípe sa dá predstaviť ako orientovaný graf, ktorého uzly sú stavy a prechody. Definuje akciu, z jedného stavu do druhého. Takto sa spočítajú nájdené riešenia a vyhodnotí sa cesta medzi počiatočným a cieľovým uzlom.

3.7 Neinformované metódy vyhľadávania

Tieto metódy vyhľadávania nemajú k dispozícii žiadne informácie o stavovom priestore. Preto musia prechádzať každý uzol systematicky, až dokým nenájdu vhodné riešenie. Medzi tieto metódy patria:

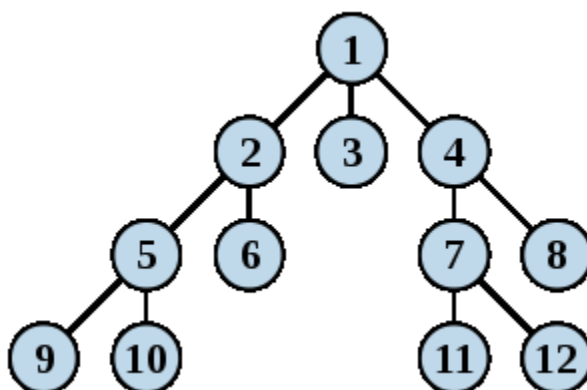
- Breadth-first search (BFS) - prehľadáva do šírky
- Depth-first search (DFS) - prehľadáva do hĺbky
- Depth-limited search (DLS) - prehľadáva do hĺbky ale s obmedzeniami
- Iterative deepening search (IDS) - interaktívne prehľadáva do hĺbky
- Bidirection search - prehľadáva obojsmerne
- Uniform-cost search - prehľadáva s uniformnou cenou

4 Výber vhodného algoritmu pre pathfinding

Pre správne vyhľadávanie cesty je potreba zvoliť vhodný algoritmus, alebo vhodnú kombináciu algoritmov. V tejto kapitole predstavím a popíšem algoritmy, ktoré som v mojej práci použil a skombinoval.

4.1 Vyhľadávací algoritmus depth-first

Algoritmus depth-first search[8], teda prehľadávanie do šírky je grafový algoritmus, ktorý postupne prechádza všetky vrcholy. Nepoužíva pritom žiadnu heuristickú analýzu. Jednoducho kontroluje všetky uzly a pre každý uzol skontroluje všetkých jeho potomkov. Počas tohto procesu si zaznamenáva rodičov jednotlivých uzlov a tým je vytvorený strom z najkratších ciest k jednotlivým uzlom z koreňa(začiatočný uzol).



Obrázok 5: Poradie v akom je pristupovane k vrcholom[5]

Z hľadiska algoritmu, každý potomok uzlu získaného expandujúcim uzlom je vkladáný do fronty typu FIFO. V tomto prípade to znamená, že prvý uzol, ktorý do fronty vstupuje je taktiež prvým uzlom ktorý ju opustí. Uzly ktoré sa dostávajú do fronty, a sú teda kontrolované, nesú označenie OPEN, uzly ktoré z fronty už vystúpili a už sa s nimi nepracuje sú označené ako CLOSE. Takéto uzly(CLOSE) už nikdy v priebehu algoritmu nebudú preskúmané, majú vyplnené všetky informácie ako sú vzdialenosť od koreňového uzlu, stav uzlu a rodiča.

4.2 Vyhľadávací algoritmus A*

Algoritmus A-Star[9] je grafový algoritmus slúžiaci k vyhľadaniu najkratšej cesty v ohodnotenom grafe vedúci zo zadaného počiatočného uzlu do cieľového. Prvýkrát ho v roku 1968 popísali P. Hart, N. Nilsson a B. Rapheal. Jeho vstupom je ohodnotený graf, počiatočný uzol a cieľový uzol, výstupom je najkratšia cesta z počiatočného uzlu do cieľového, alebo správa o tom, že žiadna cesta neexistuje.

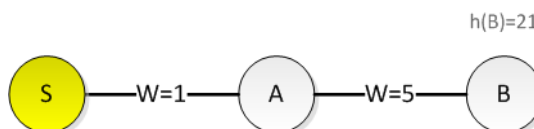
Algoritmus je z princípu zhodný s algoritmom prehľadávania do šírky, s tým rozdielom, že namiesto obyčajnej fronty používa frontu prioritnú, v ktorej sú cesty usporiadané podľa hodnoty špeciálnej funkcie **f**. Táto funkcia je definovaná pre každú cestu **p** a je súčtom heuristickej funkcie **h**, posledného uzlu cesty **p** a jej dĺžky **g**. Čím je hodnota funkcie **f(p)** nižšia, tým vyššiu má daná cesta **p** prioritu. Algoritmus sa pozerá do minulosti, čiže ako ďaleko musel zísť, dokým došiel na koniec cesty **p** i do budúcnosti, čiže ako ďaleko ešte zhruba ostáva zísť z posledného uzlu cesty **p** do cieľa.

poradie ciest vo fronte je určené funkciou : **f(x) = h(x) + g(x)**

f(x) - predpokladaná dĺžka cesty **x**

h(x) - hodnota heuristickej funkcie pre koncový uzol **x**

g(x) - dĺžka cesty **x**



Obrázok 6: Predpokladaná dĺžka cesty $(1+5)+21 = 27$

Heuristická funkcia musí splňovať dôležité podmienky - musí byť väčšia než nula a tzv. "prípustná". To znamená, že jej hodnota pre ľubovoľný uzol musí byť nižšia alebo rovná skutočnej vzdialenosti z daného uzlu do cieľa. Jej hodnota teda nikdy nemôže byť väčšia, než je skutočná vzdialenosť z daného uzlu do cieľa.

$$h(x) \leq h(y) + \text{cost}(x,y)$$

Heuristická funkcia vzniká na základe znalosti štruktúry problému. Ak sa hľadá teda napríklad najkratšia cesta z mesta **S** do mesta **G**, ide heuristickú funkciu miesta **X** použiť ostávajúcu vzdialenosť z miesta **X** do miesta **G**. V tomto prípade bude hodnota heuristickej funkcie približne rovná skutočnej hodnote.

5 Výber vhodného algoritmu pre kooperáciu

V tejto kapitole popisujem vybrané algoritmy vhodné pre kooperatívne chovanie.

5.1 Predator-prey

Algoritmus Predator-prey[10] kladie dôraz na priestorovú dynamiku interakcie dravec, korisť v 2D stavovom priestore. Počty koristi a dravcov sú náhodne generované a umiestňované na mapu.



Obrázok 7: Obrázok popisujúci pozície koristi a predátorov[9]

Ako je znázornené na obrázku, je začiatkové množstvo koristi často omnoho vyššie ako počet predátorov. Každá korisť predstavuje možný cieľ pre predátora ktorý si ale vyberá obeť podľa hodnoty jeho zdravia a umiestnenia na mape.

5.2 Pursuit-evasion

Algoritmus, ktorého princíp je založený na probléme, kde jedna skupina sa pokúša vypátrať druhú v stavovom priestore.

Existuje nespočetné množstvo možných variantov pursuit-evasion[11]. Najtypickejším prípadom je nasledujúci (policajt a zlodej): Policajti a zloději zaberajú uzly grafu(pozície v priestore). Majú na výber z možností, buď ostanú stáť na mieste a kontrolovať okolité uzly, alebo sa budú pohybovať pozdĺž okraja susedného uzlu. Ak policajt zaberie rovnaký uzol ako zlodej, tak je zlodej

zajatý a odstránený z grafu. Najčastejšia otázka v tomto prípade znie, koľko policajtov je nevyhnutných na zabezpečenie priestoru a prípadné chytenie zlodēja.

Pravidla pohybu policajtov závisia od zmeny rýchlosti zlodějov. Tato rýchlosť zodpovedá maximu počtu hrán, ktorý je zloděj schopný prejsť pozdĺžne na jeden ťah. Ak sa vo vyššie uvedenom prípade vyskytuje viacero zlodějov, majú každý rovnakú rýchlosť pohybu. V extrémnom prípade môže mať každý zloděj odlišnú rýchlosť a to teda umožňuje niektorým zlodějom rýchlejšie prechádzať grafom. Ďalšou možnosťou je udeliť policajtom schopnosť preskakovať prekážky čo znamená jednoduchší, priamočiary pohyb po mape.

Zložitosť tohto algoritmu úmerne rastie s počtom policajtov, ktorí sú potrební na vyčistenie priestoru od zlodějov. Z toho vyplýva, že pri implementácii tohto algoritmu si treba dať pozor nielen na počet zlodějov a policajtov, ale taktiež aj na veľkosť a zložitosť stavového priestoru.

6 Analýza

V prípade vývoja hry pre tak rýchlo rozvíjajúcu sa platformu akou Android je, sa hodí mať prehľad o tom, ako vyzerajú momentálne tie najpopulárnejšie hry. Netreba však zabúdať na to, že hlavným prvkom je originalita a nevšednosť. Väčšinou sa preto nevyplatí prevrat celkový koncept hry, ale posnažiť sa priniesť niečo nové.

Mimo samostatný princíp hry je vhodné sa inšpirovať aj napríklad v rozložení menu a podobne. Na takéto menu sú hráči zvyknutí a o to rýchlejšie sa v ňom vedia zorientovať. Každopádne inovácie sú aj v tomto smere vítané a hlavne nedá sa použiť celkové rozloženie menu z jednej hry na inú, pretože sa hry dosť často líšia už len v primárnych nastaveniach.

Čo sa týka grafiky, súčasný trend je taký, že by nemala nijako hráča obmedzovať a mala by byť vytvorená priamo na hru. Určite to ide ale aj inak a je dosť časté, že po jednej hre s kvalitnou grafikou vznikne niekoľko graficky veľmi podobných hier, ktoré sa taktiež stanú úspešnými. Pre príklad uvediem hru Angry Birds od spoločnosti Rovio. V danej vizuálnej stránke hra určite nebola jedinečná, a aj tak si získala veľmi veľkú popularitu medzi hráčmi. Veľa hier sa týmto úspechom inspirovalo.

Pretože toto je vôbec moja prvá skúsenosť s vývojom hry zvolil som po vizuálnej stránke veľmi jednoduchú grafiku. V práci som sa teda zamerlal na hľadanie cesty(pathfinding).

6.1 Základný koncept

Keďže moja hra je zo žánru arkády, nemusel som sa venovať zložitému príbehu. Hra by mala byť rozdelená do úrovní, kde každá by mala predstavovať zložitejšiu mapu a prípadne viacej nepriateľov, ktorých je treba zničiť. Nepriatelia sú ovládaný počítačom a teda majú vlastnú inteligenciu. Ich úkolom je vypátrať hráča na mape a zničiť ho.

Hráč pohybuje hrdinom po mape kde prechádza priestorom a snaží sa nájsť výstup do ďalšej časti mapy resp. do ďalšej úrovne. V tom sa mu snažia zabrániť nepriatelia ktorý prehľadávať priestor mapy alebo stoja a v prípade interakcie s hracom sa mu snažia zabrániť ďalej v pohybe. Hráč ma možnosť sa brániť strieľaním projektilov ktoré po zasiahnutí odstránia nepriateľa z mapy.

Cieľom hry je teda dostať hrdinu do ďalšej úrovne pričom sa musí brániť nepriateľom.

6.2 Možnosti ovládania

Ovládanie v hrách je z pohľadu hráča veľmi dôležité. V súčasnosti prevažná väčšina telefónov nie je vybavená hardwarovou klávesnicou, čiže prípadne ovládanie pomocou nej môžeme hneď na začiatku vylúčiť.

6.2.1 Ovládanie pomocou naklápania

Ovládanie v hrách pomocou naklápania funguje pomocou akcelerometru alebo gyroskopu. Tieto zariadenia slúžia na zistenie orientácie za pomoci zemskej gravitácie. Výrobcovia mobilných telefónov dosť často používajú kombináciu týchto dvoch zariadení.



Obrázok 8: Ovládanie telefónu pomocou naklápania v smere šípok na obrázku

Tento druh ovládania sa najčastejšie vyskytuje v závodných hrách kde naklápaním určujeme smer vozidla(zatáčanie). Pre moju tému arkádovej hry by takýto typ ovládania bolo neefektívne.

6.2.2 Ovládanie pomocou dotyku

Aktuálne najčastejší typ ovládania v hrách je pomocou dotyku na displej telefónu. Existujú rôzne typy ovládania, kde hráč stláča tlačidlá zobrazené na obrazovke, pohybuje virtuálnym joystickom alebo vykonáva na displeji gestá.

6.2.3 Ovládanie externým ovládačom

Platforma Android podporuje ovládanie hier aj externými ovládačmi. Tieto zariadenia sa podobajú na rovnaké zariadenia aké používajú herne konzole ako sú napr. Xbox, PlayStation. Android už od verzie 3.2 Honeycomb podporuje v základe 99% USB gamepadov, takýto gamepad stačí pripojiť a systém ho rozpozná a ovládač je pripravený na použitie.

Hlavné využitie takýchto ovládačov je pri hrách, ktoré si nevystačia s jednoduchým ovládaním. Pre príklad uvediem hru Grand Thief Auto kde pri použití gamepadu je hra plne hrateľná na rozdiel od klasického dotykového ovládania.

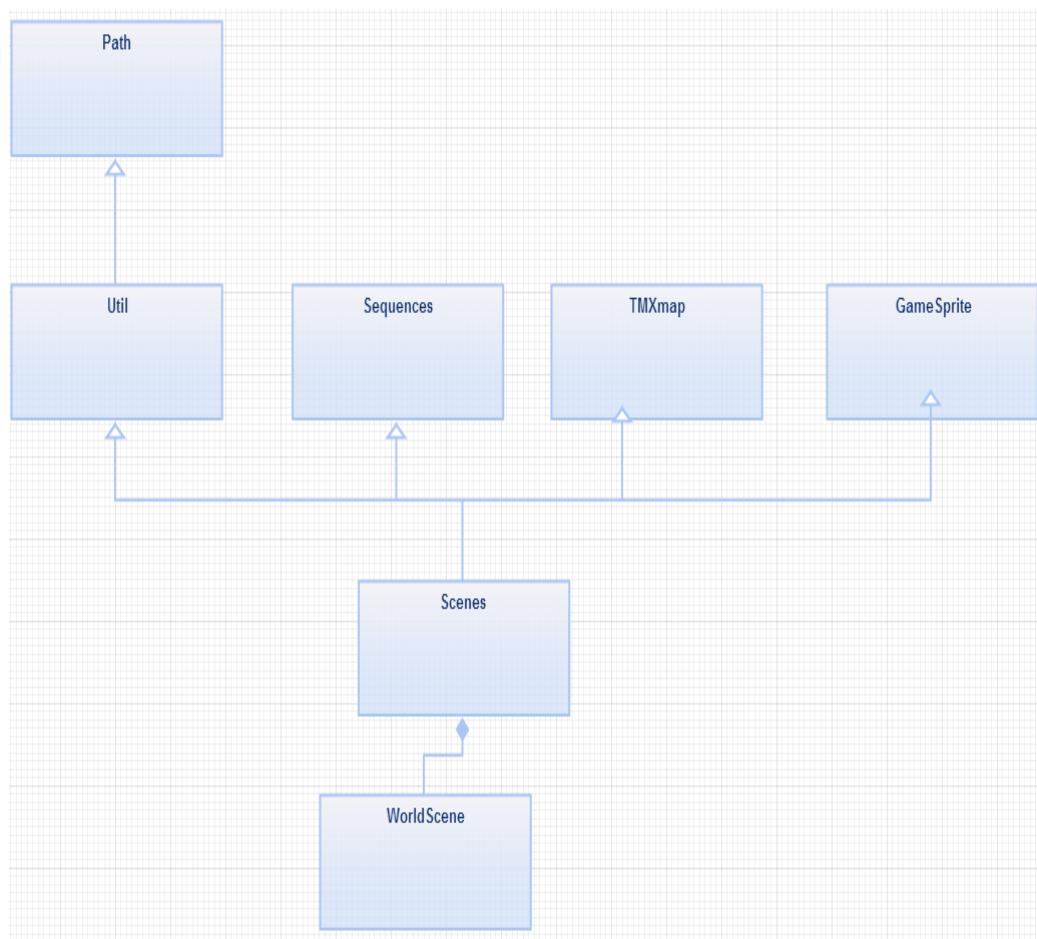


Obrázok 9: Ovládač pre telefón Samsung Galaxy Note 2

6.3 Návrh herných modulov

Hra je tvorená zo základných prvkov-modulov. Tieto moduly sú doplnené triedami, ktoré reprezentujú rôzne časti či už v užívateľskom rozhraní alebo iných funkcionalít.

Základom je modul Scenes ktorý zobrazuje obrazovku hry a komunikuje s ostatnými modulmi. Ďalšie moduly predstavujú časti ako sú vykresľovanie textúr, vykresľovanie máp, rôzne použité sekvencie a výpočet cesty(pathfinding).

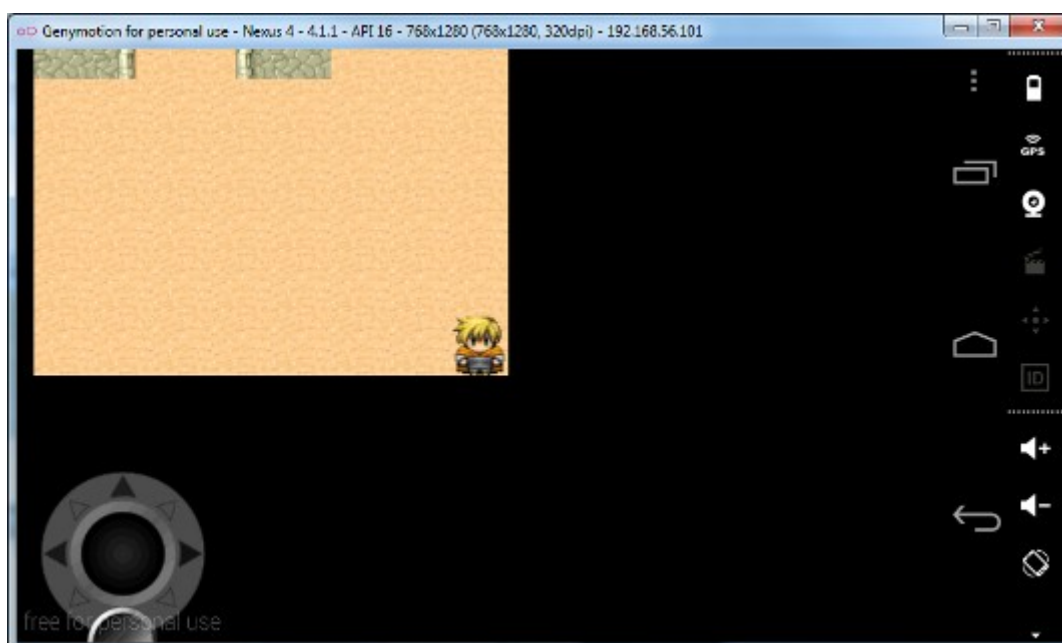


Obrázok 9: Návrh herných modulov

7 Implementácia

7.1 Ovládanie

S ohľadom na súčasnosť kde je drvivá väčšina telefónov dotykových, treba správne zvoliť ovládacie prvky. Pokiaľ má hra zaujať, musí mať v princípe jednoduché ovládanie, ktoré by sa nemalo líšiť od ostatných aplikácií. Užívatelia si tak nemusia zvykať na nové ovládanie a prakticky hneď po prvom spustení ju vedú intuitívne ovládať. Preto je nutné užívateľské rozhranie okresať na čo najmenej ovládacích prvkov.



Obrázok 10: Ukážka ovládania prostredníctvom dotykového ovládača

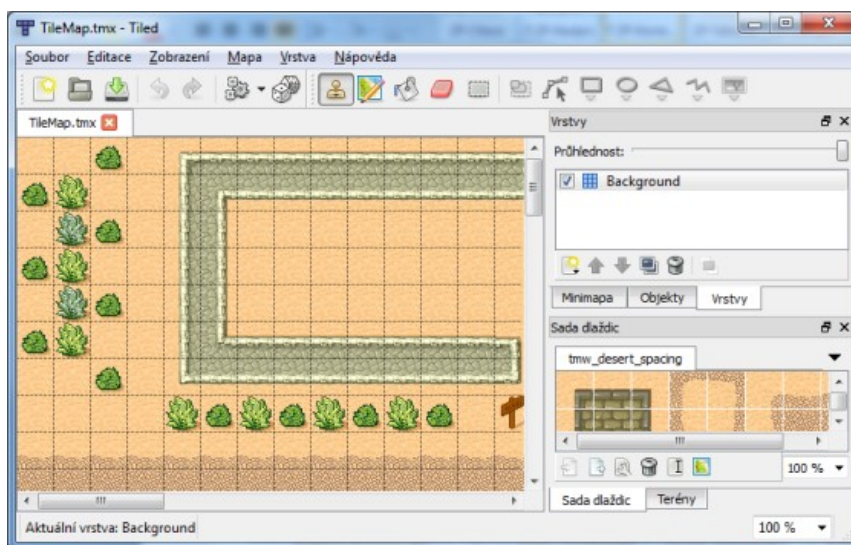
V mojom prípade som teda zvolil jednoduchý joystick ovládaný dotykom s možnosťou pohybu na 4 svetové strany. Strieľanie alebo prípadne zbieranie predmetov sa realizuje dotykom na požadované miesto.

7.2 Hlavná ponuka

Úvodná ponuka slúži pre užívateľský príjemnejšie spustenie aplikácie. Keďže štart aplikácie trvá aj niekoľko sekúnd (závisí od použitého zariadenia), dá sa tento čas vhodne využiť a zobrazíť logo aplikácie poprípade logo štúdia a podobne. Na toto som využil tzv. "splash" aktivitu. Zobrazuje sa ako prvá pri spustení a vykresľuje jednoduchý obrázok. Po vykreslení sa obrazovka presunie do hlavného menu. V hlavom menu sa nachádza pár jednoduchých nastavení. Táto časť by sa zrejme dala vylepšiť či už po grafickej stránke alebo rozdelením do viacerých pod kategórii. V mojom prípade som zvolil jednoduchosť a teda na jednej obrazovke sú všetky dostupné nastavenia.

7.3 Mapa

Stavový priestor alebo teda mapa je jeden z hlavných prvkov hry. Skladá sa z entít. Entity v tomto prípade predstavujú 2D objekty, ktoré vychádzajú zo štvorca (Rectangle) a je popísaný svojou šírkou, výškou, pozíciou a prípadne uhlom natočenia. Pre vytvorenie mapy som použil program Tiled. Umožňuje jednoduchú editáciu stavového priestoru, kde ku každej dlaždici môžeme pridať vlastnosť. Tak isto tu ide vytvárať aj vrstvy mapy. Program nám vytvorí súbor s koncovkou .tmx s ktorým neskôr pracujeme.



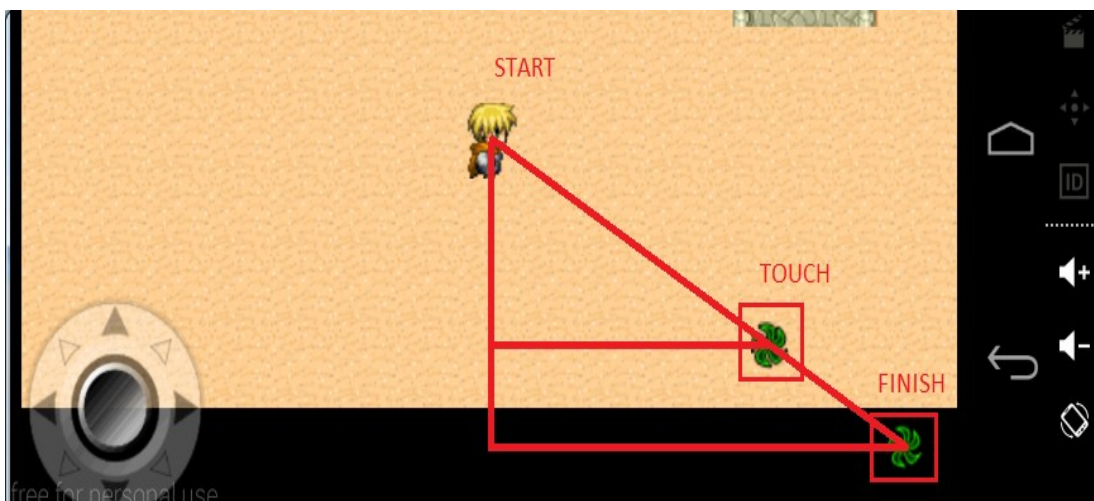
Obrázok 11: Ukážka mapy v programe Tiled

Pri načítaní mapy v programe je nutné riešiť názov mapy a jej vlastnosti. Toto všetko rieši trieda `TMXmapLoader.java`. Najpodstatnejšou časťou mapy pre moju hru sú dlaždice s vlastnosťou "COLLIDE", ako už názov napovedá jedna sa o dlaždice ktoré predstavujú pevné objekty, cez ktoré sa nedá prejsť alebo ich preskočiť. Ďalej mapa obsahuje dlaždice ktoré majú vlastnosť "ANIMATE" slúži pre jednoduché animácie ako sú pohyb trávy vo vetre a podobne. Ďalšou vlastnosťou je "EXIT" slúžia pre vstúpenie do iných častí mapy. Všetky tieto dlaždice sa vkladajú do listov. Dlaždice s vlastnosťou "COLLIDE", "EXIT" upravujú stavový priestor tak že obmedzujú pohyb hráča a nepriateľov ktorý si musia hľadať iné cesty k požadovanému cieľu.

7.4 Pohyb

Aby sa objekt mohol po mape pohybovať, je nutné meniť jeho X,Y súradnice. Tieto zmeny súradníc určujú trajektóriu, po ktorej sa objekt pohybuje. V hre sa pohybuje viacero objektov. Buď sa jedná o strely alebo aktívny pohyb nepriateľov alebo hráča. Trajektórie sa môžu líšiť. Pohyb hráča je najjednoduchší, rieši sa pohybom na 4 svetové strany vždy o jednu dlaždicu v danom smere. Pohyb nepriateľa už je zložitejší, kde na vybratú pozíciu musí zvoliť resp., vypočítať správnu cestu v tomto prípade najkratšiu. Trajektória u striel je daná začiatkom t. j. miesto, kde sa aktuálne hráč nachádza

a smerom, ktorý určí hráč dotykom na displej. Keďže je tomto prípade nevhodné poslať streľu len na miesto dotyku je potrebné vypočítať trajektóriu až po koniec mapy alebo prípadne po kolíziu s objektom takto určeným.



Obrázok 12: Ukážka trajektórie projektilu

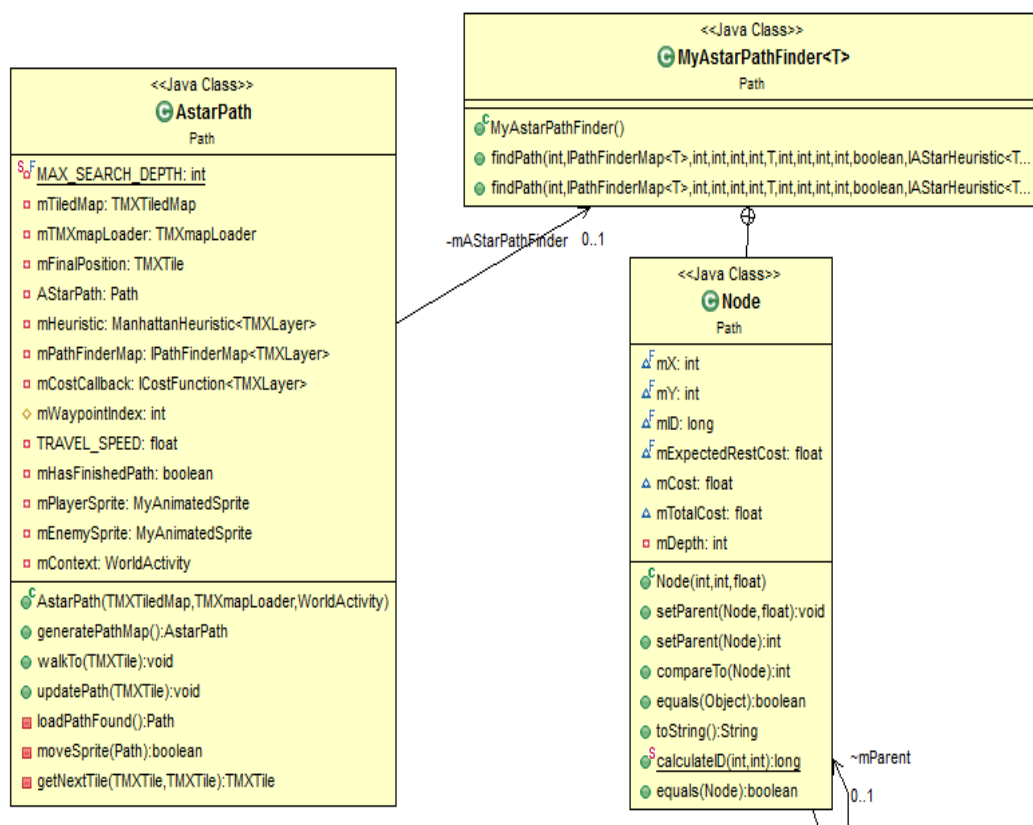
7.5 Testovanie kolízií

Vzhľadom k tomu, že v hre sa neustále pohybujú entity, tak je dôležité zisťovať, či do seba nenarazili a prípadne na tento stav reagovať.

Keďže každá entita predstavuje štvorec, tak pre testovanie kolízie stačí preveriť, či niektorý z rohových bodov entity je vo vnútri štvorca druhej entity. Tento test sa prevádza na každú dvojicu z kolekcie entít, ktoré majú pridelené vlastnosti s možnosťou kolízie. Každá takáto entita predstavuje fyzický objekt v hre a tak ako aj v reálnom svete by mala pohltiť určitú časť škôd. Konkrétne v hre využívame testovanie kolízií medzi strelami hráča a nepriateľov.

7.6 Vyhľadávanie cesty

Správne vyhľadanie cesty som realizoval kombináciou dvoch algoritmov a to konkrétne A-star, depth-first. Zložitosť tejto časti hry spočíva v množstve možností, ktoré algoritmus môže vykonať. Princíp spočíva v nájdení správnej cesty v stavovom priestore kde najskôr využijem funkciu depth-first algoritmu, ktorou nájdem možné najkratšie cesty. Tieto výsledky potom spracujem algoritmom A-star ktorý vyberie pravé jednu cestu. Toto riešenie optimalizuje prípadné chyby a prakticky úplne odstraňuje spomalenie hry pri prípadnom prepočítavaní dlhej alebo veľmi zložitej cesty.



Obrázok 13: UML diagram tried modulu pre vyhľadávanie cesty

Pseudokód:

Inicializácia: pridanie štartovacej pozície do otvoreného listu a vyprázdniť zatvorený list

Zatiaľ čo(while) sa nenašla cesta:

1.Vybrat najvhodnejší krok založený na výsledku z heuristiky a depth-first

2.Zmazat z krok otvoreného listu a pridať do uzavretého

3.Pre(for) každého suseda:

1.Vypocet ceny cesty ku susedovi

2.Ak je cena nižšia ako náklady -> vymazať z otvoreného listu alebo zavretého listu

3.Ak sa pozícia nenachádza v otvorenom alebo zavretom liste -> zapísať do otvoreného listu(brané do úvahy v ďalšom hľadaní)

Slučka končí, v prípade ak nájdeme vhodnú cestu alebo nám dôjdu možné kroky. Ak je cesta nejdená môžeme takto sledovať uzly v liste a dostať sa k požadovanému miestu.

7.7 Nastavovanie schopností jednotiek

Nastavovanie schopnosti jednotiek je možné pomocou úpravy maximálneho počtu krokov depth-first search algoritmu. Toto obmedzenie je vhodné pre reguláciu "inteligencie" nepriateľov a to v zmysle maximálnej dĺžky alebo zložitosti trasy ktoré sú schopný prejsť. Tento parameter je možné nastaviť podľa okolností. V hre sa to prejavuje tak že jednotka ktorá má príliš dlhú alebo zložitú trasu ignoruje hráča.

8 Testovanie

Zo začiatku vývoja hry bolo testovanie vykonávané na štandardnom emulátore, ktorý je súčasť balíku Android SDK. Tento emulátor absolútne nespĺňal požiadavky na testovanie a preto som musel zvoliť alternatívny emulátor Genymotion.



Obrázok 14: Spustená aplikácia v prostredí emulátoru Genymotion

Postupom času ako boli do aplikácie pridávané funkcie a možnosti, ktoré ma hra podporovať, bolo zrejmé, že testovanie na emulátore nebude dostačujúce. Preto som sa rozhodol aplikáciu otestovať v reálnych podmienkach na existujúcom mobilnom telefóne.

8.1 Testovanie na reálnom zariadení

V čase keď sa aplikácia blížila k finálnej verzii, došlo k testovaniu hry na reálnom mobilnom zariadení. K testovaciemu účelu bol použitý telefón Samsung Galaxy S4 s verziou operačného systému Google Android 4.3 (Jelly Bean) a rozlíšením displeja 1080 × 1920 bodov.

V rámci testovania bolo odohraných niekoľko hier podľa scenáru tak, aby boli vyskúšané všetky možnosti. Konkrétne sú to tieto možnosti:

- Správny pohyb hrdinu po mape
- Strieľanie projektilov, kolízia objektov
- Prechod z úrovne do úrovne
- Vyhľadávanie cesty nepriateľov
- Prispôsobovanie sa nepriateľov v rámci zložitosti a vzdialenosti cesty

Záver

Cieľom mojej bakalárske práce bolo vytvoriť 2D arkádovú hru pre platformu Android. Snažil som sa pracú napísať od obecného zoznámenia s použitými technológiami, cez voľbu algoritmov, návrh aplikácie, až po jej implementáciu. Čitateľ mojej práce sa tak postupne zoznamuje s platformou a s hrami vo všeobecnosti.

V úvode som sa zaoberal predstavením technológií ktoré som v práci použil a popisu ich hlavných častí. Ďalej som objasnil problematiku hier ako takých a zameral som sa konkrétne na arkádový žáner. V nasledujúcej kapitole som sa venoval predstavení umelej inteligencie. Popísal som jej základné časti a upriamil som pozornosť na vyhľadávanie cesty(pathfinding). Ďalej bolo potreba zvoliť správne algoritmy či už pre vyhľadávanie cesty alebo pre kooperatívne správanie jednotiek.

V nasledujúcich kapitolách som sa zameral na analýzu, návrh a samotnú implementáciu aplikácie. Popísal som tu možnosti ovládania a zvolil najvhodnejší typ pre moju aplikáciu. Vybral som z môjho pohľadu správnu kombináciu algoritmov.

Vývoj tejto hry mi umožnil zlepšiť sa v programovacom jazyku Java a veľkým prínosom bolo nadobudnutie vedomostí v oblasti tvorby aplikácií pre platformu Android. V priebehu implementácie som sa stretol z veľkým počtom problémov, o ktorých som veľa nevedel a niektoré sa mi úspešne podarili vyriešiť.

Hru sa žiaľ nepodarilo dokončiť do takej podoby kde by bola schopná uspieť aj medzi hráčmi a teda som ju neumiestnil na Google Play. V práci slúži hra ako ukážka implementovaných algoritmov ktoré sú jej hlavnou podstatou.

O platformu Android sa zaujímam a vo svojom voľnom čase by som rád pracú dopracoval do formy kde by mohla byť úspešne prezentovaná a ponúknutá širokému hráčskemu publiku na Google Play.

Literatúra

- [1] ANDROID INC. [online]. [cit. 2014-03-06]. Dostupné z: <http://developer.android.com/index.html>
- [2] Getting Started with AndEngine. AndEngine Guides [online]. 8.9.2011 [cit. 2014-02-28]. Dostupné z: <http://andengineguides.wordpress.com/2011/09/08/getting-started-with-andengine/>
- [3] DELFIN, Ricardo. AndEngine 1: Introduction to the AndEngine. IBM: FRC Tamán Keet Blog [online]. 29.1.2013 [cit. 2014-03-14]. Dostupné z: https://www.ibm.com/developerworks/mydeveloperworks/blogs/TamanKeet/entry/andengine_1_introduction_to_the_andengine_android_videogame_libraries24?lang=en
- [4] DOSTÁL, Jiří. INSTRUCTIONAL SOFTWARE AND COMPUTER GAMES – TOOLS OF MODERN EDUCATION. 2009. ISSN 1803-537X.
- [5] GOOGLE PLAY. [online]. [cit. 2014-03-06]. Dostupné z: <https://play.google.com/store>
- [6] MAŘÍK, Vladimír. Umělá inteligence. 1. vyd. Praha: Academia, 1993, 264 s. ISBN 80-200-0496-3.
- [7] NÁVRAT, Pavol. Umelá inteligencia. Bratislava: Slovenská technická univerzita, 2006, 393 s. ISBN 80-227-2354-1.
- [8] Graph Traversal, opendatastructures [online]. [cit. 2014-04-30]. Dostupné z: http://opendatastructures.org/versions/edition-0.1e/ods-java/12_3_Graph_Traversal.html
- [9] HORDĚJČUK, Vojtěch, Algoritmus A-Star [online] [cit. 2014-04-30]. Dostupné z: <http://voho.cz/wiki/informatika/algoritmus/graf/a-star/>
- [10] XIAODONG, Li. Predator-Prey Genetic Algorithm for Multiobjective Optimization. [online]. [cit. 2014-05-01]. Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.308.7463&rep=rep1&type=pdf>
- [11] VIDAL, René, Probabilistic Pursuit–Evasion Games [online]. 5.10.2002 [cit. 2014-04-30]. Dostupné z: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1067989>

Obsah priloženého CD

1. Text bakalárskej práce vo formáte PDF
2. Zdrojový súbor projektu hry v Eclipse
3. Spustiteľný súbor .APK